Analyzing closed frequent itemsets with convex polyhedra

Natalia Vanetik

Department of Software Engineering, Sami Shamoon College of Engineering,
Beer Sheva, Israel
e-mail: *natalyav@sce.ac.il*

Abstract. Computing frequent itemsets in transactional databases is a vital but computationally expensive task. Measuring the difference of two datasets is often done by computing their respective frequent itemsets despite high computational cost. This paper proposes a linear programming-based approach to this problem and shows that there exists a distance measure for transactional database that relies on closed frequent itemsets but does not require their generation.

Keywords: frequent itemsets, distance, linear programming model

1 Introduction

Suppose that we are given a dataset that consists of transactions (tuples) each containing one or more items. Frequent itemsets are subsets that appear in a large fraction of dataset tuples, where the exact fraction value is defined by the user and is called support. Frequent pattern mining was proposed by Agrawal [Agrawal,Srikant 1994] for shopping basket analysis; both frequent itemsets and association rules were introduced in this paper. Many additional algorithms have been proposed other the years, such as FPGrowth [Han, Pei, Yin 2000], Eclat [Zaki 2000], Genmax [Gouda,Zaki 2005] and others. This problem has numerous applications in both theoretical and practical knowledge discovery, but its computational complexity is another matter. It has been shown that the easier problem of counting maximal frequent itemsets, i.e. itemsets that are not a subset of other frequent itemsets, is #P-complete (see [Yang 2004]).

In this paper, we propose to use a linear programming approach for obtaining answers to some questions in frequent itemset mining without computing frequent itemsets. We introduce a proper distance measure that reflects the difference between two datasets by comparing their closed frequent itemsets; this measure is also computable in polynomial time and allows to determine whether or not the changes in a dataset over time modified its frequent itemsets.

This paper is organized as follows. Section 2 formally defines the problem of frequent itemset mining and outlines the questions we further address. Section 3 introduces the convex polyhedron model for transactional datasets and proves

its correctness. Section 4 explains how polyhedron models of two datasets can be used in order to find the distance between them. Finally, Section 5 extends the polyhedron model to the case of multiple copies of items in transactions.

2 Problem statement

Let D be a dataset of *size* m, composed of m transactions $\{t_1, \ldots, t_m\}$. Each transaction contains items that form a subset of some finite set V. The size $n := |V|$ is called the *cardinality* of D. The number of items may vary from transaction to transaction, but the items in each transaction form a set, i.e. they cannot appear more than once. Additionally, we have

$$\text{support value } 1 \leq S \leq m.$$

A set I of items (*itemset*) is called *frequent* if it appears in at least S transactions as a subset. A frequent itemset I is *closed* if no frequent set containing I has the same support. A frequent itemset I is *maximal* if no itemset containing it is frequent. Trivially, every maximal frequent itemset is closed and frequent, and every closed frequent itemset is frequent, but the inverse is not always true.

The basic task in this setting is

to find (all, closed, maximal) frequent itemsets in a given dataset.

Three main approaches to this problem are Apriori [Agrawal,Srikant 1994], FP-Growth [Han, Pei, Yin 2000] and Eclat [Zaki 2000]. This task is computationally expensive as the number of frequent itemsets in D can be exponential in dataset cardinality. Complexity of this task is preserved even if we demand frequent itemsets to be closed or maximal.

The focus of this paper is frequent and closed frequent itemsets of D. Since enumerating all frequent itemsets or even all closed frequent itemsets is hard, we ask other, less specific questions, about the set.

1. Given two datasets D_1 and D_2, how big is the difference between their respective sets CF_1 and CF_2 of closed frequent itemsets?
2. Does changing support modify the set CF, and if so, to what extent?

3 Polyhedron representation

3.1 Binary dataset representation

Let D be a dataset containing transactions $\{t_1, \ldots, t_m\}$, where each transaction t_i contains a subset of items from the set $V = \{v_1, \ldots, v_n\}$, represented by a binary matrix $M = (m_{ij})$ where $m_{ij} = 1$ if and only if $v_j \in t_i$. Every j-th column of the matrix M is a binary vector that represents all occurrences of item v_j in D. Every i-th row of M is a binary vector that represents transaction t_j of D.

$$D = \begin{array}{|l|l|} \hline \text{transaction \#} & \text{items} \\ \hline 1 & \text{bread,milk} \\ 2 & \text{milk, butter} \\ \hline \end{array}$$

$$V = \{\text{bread}, \text{milk}, \text{butter}\}$$

$$M = \begin{bmatrix} 1\ 1\ 0 \\ 0\ 1\ 1 \end{bmatrix}$$

Fig. 1. Binary matrix format of the dataset D.

$$\overline{M} = \left[\begin{array}{ccc|cc} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Fig. 2. Inverse binary matrix of the dataset D.

Example 1 *Figure 1 shows a binary format of a dataset with two transactions and three items and its matrix M where the items are numbered as $v_1 =$"bread", $v_2 =$"milk" and $v_3 =$"butter".*

We assume that

$$\textit{no item is present in all transactions.} \tag{1}$$

Note that if there are items contained in all transactions of D, we simply remove them from the dataset. Every frequent itemset in the modified dataset is transformed into a frequent itemset in the original dataset simply by adding missing items.

3.2 Inverse binary dataset representation

We use inverse $\overline{M}$ of a binary matrix M of a dataset D adjuncted to identity matrix I from the right. The purpose of identity matrix is add "counting" elements to transactions, so that each set of transactions (i.e. set of rows in $\overline{M}$) is uniquely represented by the corresponding rows of I. Using inverse matrix as opposed to the original binary matrix allows us to view transactions sets as binary sum of corresponding matrix rows.

Example 2 *Figure 2 shows the inverse binary matrix corresponding to binary matrix of Figure 1 with identity matrix I_2 attached.*

3.3 Transactions to hyperplanes

Inverse binary representation $\overline{M}$ of a dataset allows us view every transactions as a hyperplane in $\mathbb{R}^{m+n}$ in a natural way – every column of the matrix corresponds to its own variable. First n variables, denoted by $x_1, \ldots, x_n$, describe n items, and last m variables, denoted by $y_1, \ldots, y_m$, are used for transaction counting.

$$\overline{M} = \left[\begin{array}{ccc|cc} x_1 & x_2 & x_3 & y_1 & y_2 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Fig. 3. Five variables corresponding to a 2×5 matrix of Figure 2.

Denoting i-th row of $\overline{M}$ by μ_i and variables $(x_1, \ldots, x_n, y_1, \ldots, y_m)$ by $\mathbf{x}$, we obtain equations

$$H_i := \mu_i \cdot (\mathbf{x} - \mathbf{x}_0) = 0 \tag{2}$$

for some choice of point $\mathbf{x}_0$. Here, μ_i is a normal vector of hyperplane H_i which represents a single transaction in the dataset. In our model, intersections of hyperplanes will represent transactions sets. Since we are interested in transaction sets of size at least S, additional hyperplane is defined on "counting variables" y_i.

$$U := y_1 + \cdots + y_m = S \tag{3}$$

Example 3 *Figure 3 shows variables corresponding to the inverse matrix of Figure 1.*

3.4 Polyhedron model of dataset

The model Here and further, we fix the point $\mathbf{x}_0$ by setting

$$\mathbf{x}_0 = \mathbf{1}^T \tag{4}$$

and add positivity constraints for all variables

$$1 \leq x_i,\ 0 \leq y_i \leq 1 \tag{5}$$

Observe the upper half-space of every hyperplane H_i

$$H_i^{upper} := \mu_i \cdot \mathbf{x} \geq \mu_i \cdot \mathbf{x}_0$$

and the upper half-space of the hyperplane U

$$U^{upper} := y_1 + \cdots + y_m \geq S$$

We use these half-spaces and positivity constrains on x_i, y_j to define a convex polyhedron in $\mathbf{R}^{m+n}$, denoted by P.

$$P = \begin{cases} \mu_i \cdot \mathbf{x} \geq \mu_i \cdot \mathbf{1}^T, & 1 \leq i \leq m \\ y_1 + \cdots + y_m \geq S & \\ x_i \geq 1, & i \in \{1, \ldots, n\} \\ 0 \leq y_i \leq 1, & i \in \{1, \ldots, m\} \end{cases} \tag{6}$$

Our goal is to show that this polyhedron represents closed frequent itemsets in the given dataset and to use this knowledge to answer various questions about these itemsets efficiently. Since polyhedron P is defined by a system of linear inequalities, asking a question in a form of linear or quadratic function can be solved efficiently using linear programming methods.

$$m=2, n=3, S=1$$

$$P=\begin{cases}[0,0,1,1,0]\cdot[x_1,x_2,x_3,y_1,y_2]^T = x_3+y_1 \le [0,0,1,1,0]\cdot\mathbf{1}^T = 2\\ [1,0,0,0,1]\cdot[x_1,x_2,x_3,y_1,y_2]^T = x_1+y_2 \le [1,0,0,0,1]\cdot\mathbf{1}^T = 2\\ y_1+y_2 \ge S = 1\\ 1 \le x_1,x_2,x_3\\ 0 \le y_1,y_2 \le 1\end{cases}$$

Fig. 4. System (6) for 5 variables.

Example 4 *Figure 4 shows system* (6) *for the matrix* $\overline{M}$ *of Figure 1.*

Properties In this section we show that there exists a correspondence between transaction sets of sufficient size and closed frequent itemsets in the datasets. Since every transaction set is represented in our model by an intersection of corresponding hyperplanes, we need following claims.

Claim 5 *Let* H_i *and* H_j *be two of the hyperplanes defined in* (2) *and let* $i \neq j$. *Then* $H_i \cap H_j \subset H$, *where* H *is the hyperplane with normal vector* $\mu_i + \mu_j$.

Proof. Every point that satisfies equations defining H_i and H_j also satisfies their direct sum

$$(\mu_j+\mu_i)\cdot(\mathbf{x}-\mathbf{x}_0)=0,$$

which is the equation of H. □

Claim 6 *If* $\mathbf{x}_0 = \mathbf{1}^T$, *then* $H_{ij} = H_i \cap H_j$ *is the set of points* $p=(p_1,\ldots,p_n,q_1,\ldots,q_m)$ *where*

$$p_k=\begin{cases}1,\ (\mu_i\vee\mu_j)[k]=1\\ \textit{any}, \text{ otherwise}\end{cases}$$

and $q_1,\ldots,q_m$ *assume values in* $(-\infty,+\infty)$.

The normal vector of H_{ij} *is* $\mu_i \vee \mu_j$. □

Proof. Equations defining H_i and H_j for $\mathbf{x}_0=\mathbf{1}^T$ can be satisfied as equalities only if $p_k=1$ whenever $\mu_i\vee\mu_j=1$, as $0\le p_k\le 1$ for all k. If $\mu_i=\mu_j=0$, any value of p_k in range $[1,\infty)$ satisfies both equations. Thus, point p satisfies equation

$$H := (\mu_i\vee\mu_j)\cdot\mathbf{x} = (\mu_i\vee\mu_j)\cdot\mathbf{1}^T \tag{7}$$

of a hyperplane H and we have $H_{ij}\subseteq H$.

Then next step is to show that H is contained in H_{ij}.

Let $p=(p_1,\ldots,p_n,q_1,\ldots,q_m)\in H$. Since values of $q_1,\ldots,q_n$ are arbitrary for H_{ij}, we only need to observe the values of $p_1,\ldots,p_n$. Notice that $(\mu_i\vee\mu_j)[k]p_k = (\mu_i\vee\mu_j)[k]$ in case $(\mu_i\vee\mu_j)[k]=1$ for $p_k\ge 1$ if and only if $p_k=1$. If, however, $(\mu_i\vee\mu_j)[k]=0$, any value of p_k satisfies the equation of H. Then in case $\mu_i=1$ or $\mu_j=1$, we have $p_k=1$ and thus p satisfies equations of H_i and H_j. □

Claim 7 *Vertices of P have integer coordinates.*

Proof. Let $p = (p_1, \dots, p_n, q_1, \dots, q_m)$ be a vertex of P. Note that p is a vertex if and only if it is not a linear combination of other points in P.

Assume first that p_i is not integer for some $1 \le i \le n$. For p to be a vertex of P it has to lie on the intersection of at least two hyperplanes defining P. If these hyperplanes are defined by positivity constraint only, each $p_i = 1$ and $q_i = 0, 1$; the claim follows trivially. Otherwise, p lies on some hyperplane H_j. Then definition of H_j implies that

$$p_1 + \cdots + p_n + q_1 + \cdots + q_m = \mu_j \mathbf{1}^T \in \mathbb{Z}$$

Note that equation (3) implies that $q_1 + \cdots + q_m \in \mathbb{Z}$ and thus $p_1 + \cdots + p_n \in \mathbb{Z}$ Then there exists p_k, w.l.o.g. $i < k$, such that p_k is not integer. Then points

$$q = (p_1, \dots, p_i + \varepsilon, \dots, p_k - \varepsilon, \dots, p_n, q_1, \dots, q_m)$$

and

$$q = (p_1, \dots, p_i - \varepsilon, \dots, p_k + \varepsilon, \dots, p_n, q_1, \dots, q_n)$$

also lie in P while $p = \frac{1}{2}(q + s)$ – a contradiction to p being a vertex of P.

If $q_j \notin \mathbb{Z}$ for some j, equation $y_1 + \cdots + y_m = S$ of U implies that there exists $0 < q_i < 1$, w.l.o.g. $i < j$, that is also non-integer, since S is integer. Then points

$$q = (p_1, \dots, p_n, q_1, \dots, q_i + \varepsilon, \dots, q_j - \varepsilon, \dots, q_m)$$

and

$$s = (p_1, \dots, p_n, q_1, \dots, q_i - \varepsilon, \dots, q_j + \varepsilon, \dots, q_m)$$

also lie in P while $p = \frac{1}{2}(q + s)$ – a contradiction to p being a vertex of P. □

Next, we show that the faces of P represent closed frequent itemsets CF of D.

Claim 8 *There exists a face of P corresponding to every closed frequent itemset of D, and every face of P corresponds to a closed itemset in D .*

Proof. It follows from Claim 5 that faces of P represent itemsets contained in subsets of transactions of D. Indeed, if f is a face of P that w.l.o.g. satisfies equations of $H_1, \dots, H_k$ as equalities, then the equation of $H_1 \cap \cdots \cap H_k$ satisfies k equalities $\mu_i(\mathbf{x} - \mathbf{x_0}) = 0$. In this case, a point $f \in H_1 \cap \cdots \cap H_k$ has $f[i] = 1$ whenether there

fixes ()with equality to 1 all coordinates corresponding to items in transaction subset $\{t_1, \dots, t_k\}$. Equation (3) ensures that $k \ge S$. Then a normal vector $\mathbf{f}$ of face f has $f[j] = 1$, $j \in [1, n]$, whenether there exists i such that $\mu_i[j] = 1$. Frequent itemset contained in this transaction subset is closed since any frequent itemset containing it has to be contained in a larger subset of transactions and thus has bigger support.

Let now f be a face of P. Then f satisfies some of the equations in (6) as equality and the others as sharp inequality. If equalities include one or more

equations of H_i, then subset of transactions expressed by these hyperplanes has size at least S due to (3) and therefore corresponds to the inclusion-maximal itemset contained in the transaction set. Otherwise, none of the equations (2) is satisfied as equality. Since every item is absent from some transaction, we have $x_i > 1$ for all i on f . In this case, f represents the empty itemset whose support is undetermined. More than face can correspond to the itemset because of different possible values (0 or 1) of y_i's. □

Linear transformation To solve the problem of several faces of P representing the empty itemset, we apply a linear transformation $T : \mathbb{R}^{m+n} \to \mathbb{R}^{m+1}$ to the polyhedron P, where

$$\begin{aligned} T(y_1 + \cdots + y_n) &= z, \\ T(x_i) &= x_i \end{aligned} \tag{8}$$

Linear transformation of a convex polyhedron preserves linearity and convexity (see, e.g. [Barvinok, Pommersheim 1996]), application of transformation of T to P gives as a convex polyhedron

$$L = T(P) \tag{9}$$

Corollary 9 *There exists a function from the faces of L to the closed frequent itemsets CF of T.* □

4 Dataset comparison

Suppose we are given two transactional datasets D_1 and D_2 over items $V = \{v_1, \ldots, v_m\}$ (both datasets may also represent the state of a single dataset D in different points of time). We wish to compare sets of closed frequent itemsets of D_1 and D_2, denoted by CF_1 and CF_2, for given support values S_1 and S_2, where S_1 and S_2 may differ.

A straightforward method of comparing CF_1 and CF_2 is to compute them and then calculate the distance between them according to the chosen metric, such as Hamming distance, Manhattan or Euclidean distance if the values are numerical etc. However, computing sets CF_1 and CF_2 is an expensive task, as the sizes of these sets may very well be exponential in m.

We propose another approach to solve this problem. First, we adjust the smaller of these datasets (say, D_1) so that $|D_1| = |D_2|$ by adding $|D_2| - |D_1|$ transactions containing all the items and increasing S_1 accordingly by setting $S_1 := S_1 + |D_2| - |D_1|$. Any itemset that was frequent in D_1 before the adjustment remains frequent after the adjustment as well. Now, let us observe two polyhedra L_1 and L_2 defined in (9) for datasets D_1 and D_2 respectively. In order to compensate for the difference in support values, we apply a simple linear transformation $T_S : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$T_S(z) = \frac{S_1}{S_2} z,$$

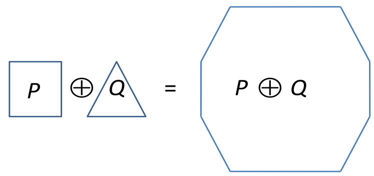

Fig. 5. Minkowski sum of two convex polygons.

where z is the variable defined in (8).

Question of distance between datasets is easily translated to the question of polyhedron distance and/or the volume of their difference. Since computing the volume of a polytope is NP-hard (see e.g. [Dyer, Frieze 1988]), we employ another kind of distance measure called *polyhedra penetration depth*.

Definition 10 *Let P and Q be two intersecting polyhedra. The penetration depth of P and Q, denoted by $PD(P,Q)$, is the minimum translational distance (translation every point a constant distance in a specified direction) that one of the polyhedra must undergo to render them disjoint.*

In other words, penetration depth $PD(P,Q)$ is a minimum distance from the difference vector between the origins of P and Q, to the surface of the Minkowski sum (denoted by $\oplus$) of P and $-Q$. An example of Minkowski sum is given in in Figure 5. The computational complexity of computing the Minkowski sum is $O(m^2)$ in case when P and Q are convex polyhedra, i.e. polynomial in m (see [O'Rourke 1994] for survey of algorithms; fastest algorithms are described in [Kaul, O'Connor, Srinivasan 1991] and [Sharir 1987]).

With this approach, we can answer affirmatively the following questions about D_1 and D_2 (and to obtain the answer in time polynomial in n, m).

1. Distance between D_1 and D_2 can be computed as $PD(T_S(L_1), T_S(L_2))$.
2. The common part of CF_1 and CF_2 is the polyhedron $T_S(L_1) \cap T_S(L_2)$. While computing the volume of $T_S(L_1) \cap T_S(L_2)$ is hard, it is easy to determine whether or not this polyhedron is empty by minimizing any suitable objective function, for instance $f = y_1$ etc. If the polyhedron is empty, the function is infeasible. Since this is an LP task, it can be done in polynomial time.
3. In a special case where $D_1 = D_2$ but w.l.o.g. $S_1 < S_2$, the polyhedron $L_1 \oplus (-L_2)$ is convex. This polyhedron is empty if the change in support value does not change frequent itemsets.

This approach to computing distance between sets of frequent itemsets can be employed in *inverse frequent itemset mining*. This task is defined as the problem

$D' =$

transaction #	items
1	bread, bread, milk, milk
2	milk, butter, butter, butter

$$\text{items } V = \{\text{bread}_1, \text{bread}_2, \text{milk}_1, \text{milk}_2, \\ \text{butter}_1, \text{butter}_2, \text{butter}_3\}$$

transaction #	items
1	$\text{bread}_1, \text{bread}_2, \text{milk}_1, \text{milk}_2$
2	$\text{milk}_1, \text{butter}_1, \text{butter}_2, \text{butter}_3$

$$M = \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0 \\ 0\,0\,1\,0\,1\,1\,1 \end{bmatrix} \qquad \overline{M} = \left[\begin{array}{c|c} 0\,0\,0\,0\,1\,1\,1 & 1\,0 \\ 1\,1\,0\,1\,0\,0\,0 & 0\,1 \end{array}\right]$$

Fig. 6. Binary representation of a generalized dataset.

of generating synthetic datasets from frequent itemsets that have been computed already. While in general this problem was shown to be NP-complete (see [Mielikäinen 2003], [Calders 2004], [Wang, Wu 2005]), the problem of finding whether or not the dataset generated already has the same set of frequent itemsets is much easier. Indeed, given dataset D and frequent itemsets F, we can construct the polyhedron L of frequent itemsets of D and compute Minkowski sum $L \oplus (-F)$ in polynomial time. If this sum is empty, D's closed frequent itemsets are entirely described by the set CF. Otherwise, $PD(L, CF)$ is the distance between them.

5 Generalized datasets

In this section, we take a look at a generalized problem of analyzing frequent itemsets in a dataset D where an item may appear more than once in a transaction.

The exact problem setting is as follows. We have a dataset $D = \{t_1, \ldots, t_m\}$ where each transaction t_i is a *multiset* of items from $V = \{v_1, \ldots, v_n\}$. For given natural $1 \leq S \leq m$, a multiset I of items is *frequent* if there are at least S instances of it in D. Note that D cannot be represented as a binary matrix directly. Instead, we apply the following transformation:

– If transaction t_i contains k copies of item v_j, we replace these copies with synthetic items $v_{j,1}, \ldots, v_{j,k}$.

In the end, we obtain a non-generalized dataset that can be represented by a binary matrix. Figure 6 show how this change in implemented on a dataset of size 2 and cardinality 3.

After this transformation, we can build system (6) of linear inequalities for the modified matrix inverse. We obtain a convex polyhedron P that represents our generalized dataset D.

Using results of Section 3, we conclude that

1. P is a convex polyhedron whose faces represent closed frequent itemsets of D.
2. Applying linear transformation $T_S(\sum_i y_i) = z$ to P results in a convex polyhedron $L(P)$ whose faces represent closed frequent itemsets of D.

Then following results hold for our model.

1. Enumerating the vertices of P is #P-hard (see [Barvinok, Pommersheim 1996]).
2. Computing the volume of P is an NP-hard problem.
3. The problem of classifying datasets up to equivalence of their frequent itemsets is wild.
4. Penetration depth of two polyhedra L_1 and L_2 can be computed in polynomial time and it is a measure of distance between the sets of frequent itemsets of two datasets or a measure of change of the same dataset over time or over different support values.

6 Conclusions

This paper proposes a linear programming approach in order to compare closed frequent itemsets of two datasets; such an approach can also be applied to the case of a single dataset that changes over time. Systems of linear inequalities are used to express closed frequent itemsets as convex polyhedra; polyhedra penetration depth measure is then used as a distance measure for the two datasets.

References

Agrawal,Srikant 1994. Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487–499, Santiago, Chile, September 1994.

Barvinok, Pommersheim 1996. A. Barvinok and J.E. Pommersheim, An algorithmic theory of lattice points in polyhedra, New Perspectives in Algebraic Combinatorics (Berkeley, CA, 1996–97), Math. Sci. Res. Inst. Publ., vol. 38, Cambridge Univ. Press, Cambridge, 1999, pp. 91–147. Appl. 361, 2003, pp. 203-222.

Buelk, Luoto 2008. 0/1 Polytopes Survey, Samuel J. Buelk, Kurt Luoto May 25, 2008.

Calders 2004. T. Calders. Computational complexity of itemset frequency satisfiability. In: Proc. 23rd ACM PODS 04, pp. 143–154, ACM Press, 2004.

Dyer, Frieze 1988. Martin E. Dyer, Alan M. Frieze, On the Complexity of Computing the Volume of a Polyhedron. SIAM Journal of Computing, volume 17(5) pp. 967–974, 1988.

Gouda,Zaki 2005. Karam Gouda and Mohammed J. Zaki, GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets. Data Mining and Knowledge Discovery: An International Journal, 11(3):223–242. Nov 2005.

Han, Pei, Yin 2000. Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD00), Dallas, TX, pp 1–12.

Kaul, O'Connor, Srinivasan 1991. A. Kaul, M. A. O'Connor, and V. Srinivasan. Computing Minkowski sums of regular polygons. In Proc. 3rd Canad. Conf. Comput. Geom., pp. 74–77, 1991.

Kuznetsov 1989. S.O. Kuznetsov, Interpretation on Graphs and Complexity Characteristics of a Search for Specific Patterns, Nauchn. Tekh. Inf., Ser. 2 (Automatic Document Math Linguist), vol. 23, no. 1,pp. 23–37, 1989.

Mielikäinen 2003. T. Mielikäinen. On inverse frequent set mining. In: Proc. of 2nd Workshop on Privacy Preserving Data Mining (PPDM), pp. 18–23, IEEE Computer Society, 2003.

O'Rourke 1994. J. O'Rourke. Computational Geometry in C. Cambridge University Press, New York, 1994.

Sharir 1987. M. Sharir. Efficient algorithms for planning purely translational collision-free motion in two and three dimensions. In Proc. IEEE Internat. Conf. Robot. Autom., pages 1326–1331, 1987.

Wang, Wu 2005. Y. Wang, X. Wu. Approximate Inverse Frequent Itemset Mining: Privacy, Complexity, and Approximation. In Proceedings of ICDM 2005.

Yang 2004. Yang, G. (2004) The complexity of mining maximal frequent itemsets and maximal frequent patterns. Proc. Int. Conf. Knowl. Discov. Data Mining, pp. 344–353.

Zaki 2000. Zaki, M.J. (2000) Scalable algorithms for association mining, IEEE TransKnowl Data Eng 12:372–390.